Amendments to the Specification:

Please replace paragraph [00013] with the following amended paragraph:

[00013] A solution for secured access by a wide range of users to company computer resources and data is by using a way to enable "remote control" of a user's application so that it may communicate and bypass a firewall to access the computer resources. FIGURE 1 is a simplified block diagram of an embodiment of a system and method 10 for remote application process control. A secure computing environment 12 is protected from unauthorized access by a firewall 14 deployed in the demilitarized zone (DMZ) at the interface between secure environment 12 and non-secure environment 22. Firewall 14 may be any hardware and/or software that prevent unauthorized access to or from a private network. Firewall 14 may be a packet filter, an application gateway, a circuit-level gateway, a proxy server, or a combination of these systems. For example, firewall 14 may be the ~~Applicatoin~~ Application Security Gateway (ASG) offered by Permeo Technologies, Inc. of Irving, Texas. ASG is a secure bi-directional proxy- based application gateway that supports Internet transport protocols such as TCP (Transmission Control Protocol) and UDP (User Datagram Protocol). In secure computing environment 12, a plurality of hardware and/or software resources 16, including a mail server ~~16~~ 18 and a web server ~~18~~ 20, for example, are accessible via firewall 14 by users external to secure environment 12. Users in non-secure environment 22 may use a wide range of computing devices 24 such as desktop computers, laptop computers, notebook computers, personal digital assistants, and devices now known or to be developed. These computing devices 24 use wired and wireless technologies and protocols to communicate with other computing devices via one or more public and/or private networks.

Please replace paragraph [00018] with the following amended paragraph:

[00018] FIGURE 5 is a more detailed flowchart of another embodiment of a method 90 for remote application control. Method 90 is a preferred mechanism used for operating systems such as WINDOWS NT, WINDOWS 2000 and WINDOWS XP. In block 90, injector 26 uses CreateProcess( ) with the CREATE_PROCESS_SUSPENDED option to launch application 30 so that it starts in a suspended state. This suspend mechanism is used to interrupt the execution of the application process. Injector 26 uses a function such as VirtualAllocEx to create memory inside the suspended application 30 in block 92 and injects redirect code 28 into the created memory space in block 94. The last instruction in redirect code 28 copied into the created memory space is a break point, for example. Injector 26 then sets the instruction pointer to the beginning of the created memory space where the first instruction of redirect code 28 is located.

in block 96 and the redirect code executes in block 98. When redirect code 28 comes to its last instruction, injector 26 catches the break point in block 100 and resets the instruction pointer to the application's main thread in block 110. The application ~~begins~~ resumes execution in block 112. Further exceptions and break points are handed off to the operating system to handle and injector 26 exits when application 30 exits in block 114.

Please replace paragraph [00021] with the following amended paragraph:

[00021] FIGURE 8 is a simplified flowchart of an embodiment of a process ~~144~~ 148 of intercepting application function calls. Process ~~155~~ 148 is performed by an import table replacement function in redirect code 28. Process ~~144~~ 148 recursively searches the import tables (in the portable executable header) of the application's main module in block 150 as well as all dynamic link libraries that are mapped into the process space in block 152 for a set of target function addresses. The target functions are functions that redirect code 26 is interested in intercepting so that its functionality may be remotely controlled. For example, all socket functions exported by WINSOCK.DLL, WSOCK32.DLL, and WS2_32.DLL are target functions when the application process control is interested in intercepting network traffic or socket calls to securely communicate via firewall 14. To accomplish other goals, other types of target functions may be intercepted using this method. When the target function addresses are found, the in-process memory of the module is modified to point at the replacement function addresses in redirect dynamic link library 36 in block 154. Because the previous steps would enable the intercept of functions in the dynamic link libraries whose import tables were linked into the application or other dynamic link libraries at compile time, process ~~155~~ 148 also ~~need~~ needs to intercept functions in the dynamic link libraries that are loaded at runtime. In blocks 156 and 158, kernel function calls such as LoadLibrary( ) and GetProcAddress( ) are also intercepted and the results of those function calls are replaced with redirect function addresses. Further, process ~~155~~ 148 also targets applications that get started by the main application, CreateProcess( ) family of functions are also of interest. To capture all of these function calls, process 155 ~~155~~ 148 recursively searches and replaces function calls of interest in those dynamic link libraries and any additional dynamic link libraries that may be addressed by them in block 158. When LoadLibrary( ) function is encountered at runtime, once the dynamic link library is loaded into memory, the import table of that dynamic link library, and recursively, any dynamic link libraries loaded by that dynamic link library are searched and the import table function replacement is carried out. When GetProcAddress( ) is encountered at runtime for a function that has been replaced, the address of the replacement function in the redirect dynamic link library is returned instead. When CreateProcess( ) or one of its variant

functions is intercepted at runtime, a new injector is started that performs the injection procedure on the new target application as described above.